REMARKS

This is a full and timely response to the outstanding non-final Office Action mailed June 1, 2007. The Examiner is thanked for the thorough examination of the present application. Upon entry of this response, claims 1, 3-10, and 12-26 are pending in the present application.

Claims 1, 4, 5, 7-9, 13-15, 17, 19-21, and 23-26 are rejected under 35 U.S.C. §102(e) as allegedly being anticipated by *Voorhies et al.* (U.S. Pat. No. 7,023,437, hereinafter "*Voorhies*"). Claims 3, 6, 10, and 18 are rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over *Voorhies* in view of *Gannett* (U.S. Pat. No. 6,118,452). Claims 12, 16, and 22 are rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Voohies in view of *Griffin* (U.S. Pat. No. 5,990,904).

Applicants respectfully request consideration of the following remarks contained herein. Reconsideration and allowance of the application and presently pending claims are respectfully requested.

RESPONSE TO CLAIM REJECTIONS UNDER 35 U.S.C. § 102

Claims 1, 4, 5, 7-9, 13-15, 17, 19-21, and 23-26 are rejected under 35 U.S.C. §102(e) as allegedly being anticipated by *Voorhies*. For at least the reasons set forth below, Applicants traverse these rejections.

Independent Claim 1

Applicants respectfully submit that independent claim 1 patently defines over *Voorhies* for at least the reason that *Voorhies* fails to disclose, teach or suggest certain features in claim 1.

Claim 1 recites (emphasis added):

> 1. A multi-pass method of rendering a plurality of graphic primitives comprising:
>
> in a first pass:
>
> passing only a limited portion of graphic data for each primitive through a graphic pipeline, wherein the limited portion of graphic data comprises location-related data;
>
> ***<u>processing the limited portion of graphic data to build a compressed z-buffer, the compressed z-buffer comprising a plurality of z-records, each z-record embodying z information for a plurality of pixels</u>***;
>
> setting a visibility indicator, for each primitive, if any pixel of the primitive is determined to be visible;
>
> in a second pass:
>
> for each primitive, determining whether the associated visibility indicator for that primitive is set;
>
> discarding, without passing through the graphic pipeline, the primitives for which the associated visibility indicator is not set;
>
> passing the remaining portion of graphic data for each primitive determined to have the associated visibility indictor set;
>
> ***<u>performing a two-level z-test on graphic data, wherein a first level of the z-test compares the graphic data of a current primitive with corresponding information in the compressed z-buffer, and wherein a second level of the z-test is performed on a per-pixel basis in a z-test manner</u>***, wherein the second level z-test is performed only on pixels within a record of the compressed z-information in which the first level z-test determines that some but not all pixels of an associated macropixel are visible; and
>
> communicating data associated with pixels of macropixels determined to be visible to a pixel shader for rendering.

Applicants respectfully submit that the *Voorhies* reference fails to teach the feature "processing the limited portion of graphic data <u>to build a compressed z-buffer</u>, the compressed z-buffer comprising a plurality of z-records, each z-record embodying z information for a plurality of pixels." In alleging that *Voorhies* teaches this feature, the Office Action refers to the following text passages from *Voorhies* (emphasis added):

> The novel algorithm presented herein involving **<u>hierarchical z-buffering is more efficient</u>** and more suitable for hardware implementation than algorithms that have been used previously. **<u>The algorithm performs z-buffer tiling hierarchically on NxN</u>**

> **regions of image space using a z-pyramid having NxN decimation from level to level to store the depths of previously rendered polygons.**
>
> At each cell encountered during hierarchical tiling of a polygon, **conservative culling is performed very efficiently by comparing the z-pyramid value to the depth of the plane of the polygon**. This routine hierarchically evaluates the line and plane equations describing a polygon using a novel algorithm that does not require general-purpose multiplication (except for set-up computations).

(Col. 6, lines 1-14).

> In use, the coarse rasterizer 3792 processes tiles at a first size (i.e. 16x16 pixel region). Such processing includes z-value culling and stencil culling in the manner set forth hereinabove. **Thereafter, the coarse rasterizer 3792 sends an output of such processing to the normal rasterizer 3794 which processes tiles at a finer size (i.e. 4x4 pixel region)**. In particular, the normal rasterizer 3794 performs the z-value culling and stencil culling on the tiles at the finer level. **By this design, tiles that are culled during processing with the coarse rasterizer 3792 may be skipped** during processing with the normal rasterizer 3794, thus saving considerable bandwidth and computation.

(Col. 54, lines 44-55). *Voorhies* teaches of performing rasterization in multiple stages. (See embodiment 3790 in FIG. 37C of *Voorhies*.) This involves a coarse rasterizer 3792 and a normal rasterizer 3794. The coarse rasterizer 3792 processes tiles at a first size (*i.e.* 16x16 pixel region). Thereafter, the coarse rasterizer 3792 sends an output of such processing to the normal rasterizer 3794 which processes tiles at a finer size (*i.e.* 4x4 pixel region). In particular, the normal rasterizer 3794 performs the z-value culling and stencil culling on the tiles at the finer level. Thus, tiles that are culled during processing with the coarse rasterizer 3792 may be skipped during processing with the normal rasterizer 3794. However, nowhere does *Voorhies* appear to even disclose processing the limited portion of graphic data to build a compressed z-buffer, the compressed z-

<u>buffer comprising a plurality of z-records</u>, each z-record embodying z information for a plurality of pixels.

At most, *Voorhies* discloses a z-pyramid, where the z-pyramid is organized preferably in NxN tiles, as illustrated in FIG. 2 of the *Voorhies* reference for a three-level pyramid 200 organized in 4x4 tiles. Each tile is a 4x4 array of "cells," which <u>are samples 202 at the finest level of the pyramid and square regions of the screen 206 at the other levels</u>. *Voorhies* further teaches that the finest level of the z-pyramid 200 is a z-buffer <u>containing the depth of the nearest primitive encountered so far at each image sample</u>, and the other levels contain zfar values, indicating the depths of the farthest depth samples in the z-buffer within the corresponding square regions of the screen. (Col. 8, lines 43-55). *Voorhies* also teaches that since a z-pyramid has a plurality of levels which are each a depth buffer, it can also be described as a hierarchical depth buffer. Applicants refer to the following text passage in the *Voorhies* reference:

> **<u>A cell in the z-pyramid is the region of the screen corresponding to a value in the z-pyramid.</u>** Preferably, **<u>at the finest level</u>** of the z-pyramid, cells correspond to depth samples--depths at pixels when point sampling and depths at subpixel samples when oversampling. **<u>At coarser levels</u>** of the z-pyramid, cells correspond to square regions of the screen, as with image pyramids in general.

(Col. 6, lines 39-46).

However, Applicants respectfully submit that the z-pyramid is not equivalent to the compressed z-buffer claimed by Applicants. As noted on page 4 of the Office Action, a compressed z-buffer is formed during the first-pass processing. The z-buffer effectively provides condensed depth information for

multiple pixels, such that a grouping of pixels (or a macro-pixel) may be trivially accepted (during the second pass) if all pixels of a current macro-pixel are deemed to be in front of previously-stored pixels or trivially rejected if all pixels of the current macro-pixel primitive are deemed to be behind previously-stored pixels. For purposes of illustration only, Applicants further refer to FIG. 3 of the present application, which is a block diagram illustrating a compression of a z-buffer. The compressed z-buffer 306 shown in FIG. 3 is just one embodiment where z-information for sixty-four pixels (an eight by eight pixel block, or macro-pixel) is compressed into a single record. Applicants respectfully submit that *Voorhies* fails to teach of building a compressed buffer.

Furthermore, Applicants submit that *Voorhies* fails to teach performing a <u>two-level z-test</u> on graphic data as recited in claim 1, wherein:

> <u>*a first level*</u> of the z-test compares the graphic data of a current primitive with
> corresponding information in the compressed z-buffer, and
> <u>*a second level*</u> of the z-test is performed on a per-pixel basis in a z-test manner,
> where <u>the second level z-test is performed only on pixels within a record of the compressed z-information in which the first level z-test determines</u> that some but not all pixels of an associated macropixel are visible.

At most, *Voorhies* teaches that "[t]he algorithm performs z-buffer tiling hierarchically on NxN regions of image space using a z-pyramid having NxN decimation <u>from level to level to store the depths of previously rendered polygons</u>." (Col. 6, lines 4-8). The "levels" disclosed in the *Voorhies* reference relate to different levels in a z-pyramid. *Voorhies* further teaches that "[s]ince a

z-pyramid has a plurality of levels which are each a depth buffer, it can also be described as a hierarchical depth buffer." (Col. 8, lines 56-58).

In contrast, the term "level" recited in claim 1 relates to differing levels of a z-test. The second level of the z-test depends in part on the outcome of the first level z-test ("wherein the second level z-test is performed only on pixels within a record of the compressed z-information in which the first level z-test determines that some but not all pixels of an associated macropixel are visible"). Applicants respectfully submit that *Voorhies* fails to teach this feature.

Accordingly, Applicants respectfully submit that independent claim 1 patently defines over *Voorhies* for at least the reason that *Voorhies* fails to disclose, teach or suggest the highlighted features in claim 1 above.

Dependent Claims 3-7

Applicants submit that dependent claims 3-7 are allowable for at least the reason that these claims depend from an allowable independent claim. *See, e.g., In re Fine*, 837 F. 2d 1071 (Fed. Cir. 1988).

Independent Claim 8

Claim 8, as amended, recites (emphasis added):

> 8. A method of rendering a plurality of graphic primitives comprising:
> passing, within a graphic pipeline, only a limited portion of the graphic data associated with each primitive, wherein the limited portion of graphic data comprises location-related data; and wherein each primitive comprises a plurality of pixels;
> ***processing the limited portion of graphic data associated with each individual primitive to build a compressed z-buffer for***

<u>each primitive, wherein each z-buffer contains compressed z-information for a macro-pixel</u>;

determining, for each primitive, whether the primitive has at least one visible pixel;

communicating data associated with pixels of primitives determined to have at least one visible primitive to a pixel shader for rendering; and

passing and processing, within the pixel shader, the remaining graphic data associated with each primitive only for those primitives determined to have at least one visible pixel, wherein the remaining graphic data includes at least one of the following: lighting, texture, and fog data.

Applicants have amended claim 8 to clarify certain novel features and submit that no new matter is added by the amendment. Applicants submit that *Voorhies* fails to teach "processing the limited portion of graphic data associated with each individual primitive to build a compressed z-buffer for each primitive, wherein each z-buffer contains compressed z-information for a macro-pixel." As discussed in depth above, Applicants submit that while *Voorhies* discloses use of a z-pyramid, this is not equivalent to the compressed z-buffer claimed by Applicants.

Accordingly, Applicants respectfully submit that independent claim 8 patently defines over *Voorhies* for at least the reason that *Voorhies* fails to disclose, teach or suggest the highlighted features in claim 8 above.

<u>Dependent Claims 9, 10, and 12</u>

Applicants submit that dependent claims 9, 10, and 12 are allowable for at least the reason that these claims depend from an allowable independent claim. *See, e.g., In re Fine*, 837 F. 2d 1071 (Fed. Cir. 1988).

<u>Independent Claim 13</u>

Claim 13 recites (emphasis added):

> 13. A method of rendering a plurality of graphic primitives comprising:
>
> passing in a first pass, within a graphic pipeline, only a limited portion of graphic data for each primitive, wherein each primitive comprises a plurality of pixels and wherein the limited portion of graphic data comprises location-related data;
>
> ***<u>processing the limited portion of graphic data to build a compressed z-buffer, the compressed z-buffer comprising a plurality of z-records, each z-record embodying z information for a plurality of pixels</u>***;
>
> in a second pass, within the graphic pipeline, ***<u>performing a two-level z-test on graphic data, wherein a first level of the z-test compares the graphic data of a current primitive with corresponding information in the compressed z-buffer, and wherein a second level of the z-test is performed on a per-pixel basis in a z-test manner, wherein the second level z-test is performed only on pixels within a record of the compressed z-information in which the first level z-test determines that some but not all pixels of a macropixel are visible</u>***, wherein additional graphic data associated with each primitive is passed into the graphics pipeline on the second pass only for primitives that are at least partially visible; and
>
> communicating data associated with pixels of macropixels determined to be visible to a pixel shader for rendering.

Applicants respectfully submit that *Voorhies* fails to teach the features emphasized above in claim 13. As discussed in depth above, Applicants submit that while *Voorhies* discloses use of a z-pyramid, this is not equivalent to the compressed z-buffer claimed by Applicants. *Voorhies* also fails to disclose the two-level z-test recited in claim 13.

Accordingly, Applicants respectfully submit that independent claim 13 patently defines over *Voorhies* for at least the reason that *Voorhies* fails to disclose, teach or suggest the highlighted features in claim 13 above.

<u>Independent Claim 14</u>

Claim 14, as amended, recites (emphasis added):

> 14. A graphics processor comprising:
> first-pass logic configured to deliver to a graphic pipeline, in a first pass, only a limited portion of graphic data for each primitive, wherein each primitive comprises a plurality of pixels, wherein the limited portion of graphic data comprises location-related data;
> ***<u>logic configured to process the limited portion of graphic data for each primitive to create a compressed z-buffer comprising a plurality of z-records, wherein z-information for a macro-block is compressed into each of the plurality of z-records</u>***;
> logic configured to determine, for each primitive, whether the primitive has at least one visible pixel;
> second-pass logic configured to deliver to the graphic pipeline, in a second pass, the remaining graphic data associated with each primitive for only those primitives determined to have at least one visible pixel, the second-pass logic further configured to inhibit the delivery of graphic data to the graphic pipeline for primitives not determined to have at least one visible pixel.

Applicants have amended claim 14 to clarify certain novel features and submit that no new matter is added by the amendment. Applicants respectfully submit that *Voorhies* fails to teach the features emphasized above in claim 14. As discussed in depth above, Applicants submit that while *Voorhies* discloses use of a z-pyramid, this is not equivalent to the compressed z-buffer claimed by Applicants.

Accordingly, Applicants respectfully submit that independent claim 14 patently defines over *Voorhies* for at least the reason that *Voorhies* fails to disclose, teach or suggest the highlighted features in claim 14 above.

<u>Dependent Claims 15-20</u>

Applicants submit that dependent claims 15-20 are allowable for at least the reason that these claims depend from an allowable independent claim. *See, e.g., In re Fine*, 837 F. 2d 1071 (Fed. Cir. 1988).

Independent Claim 21

Claim 21, as amended, recites (emphasis added):

> 21. A graphics processor comprising:
> logic configured to pass and process only a portion of graphic data passed into a graphic pipeline for each of a plurality of primitives, in a first pass within the graphic pipeline to determine whether the primitive has at least one visible pixel, wherein each primitive comprises a plurality of pixels, and wherein the limited portion of graphic data comprises location-related data;
> ***logic configured to build a compressed z-buffer from processing of the graphic data in the first pass, the z-buffer comprising a plurality of z-records, wherein z-information for a macro-block is compressed into a single record***; and
> logic configured to render, in a second pass within the graphic pipeline, only the primitives determined in the first pass to have at least one visible pixel, wherein the remaining portion of graphic data associated with each primitive is passed into the graphics pipeline on the second pass.

Applicants have amended claim 21 (and canceled claim 24) to further define certain features in claim 21. Applicants submit that no new matter is added by the amendment. Applicants submit that *Voorhies* fails to teach the compressed z-buffer disclosed in claim 21. As discussed above, while *Voorhies* teaches of utilizing z-buffer tiling hierarchically using a z-pyramid, Applicants submit that this is not equivalent to the z-buffer recited in claim 21.

Dependent Claims 22-26

Applicants submit that dependent claims 22-26 are allowable for at least the reason that these claims depend from an allowable independent claim. *See, e.g., In re Fine*, 837 F. 2d 1071 (Fed. Cir. 1988).

CONCLUSION

Applicants respectfully submit that all pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephone conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

No fee is believed to be due in connection with this amendment and response to Office Action. If, however, any fee is believed to be due, you are hereby authorized to charge any such fee to deposit account No. 20-0778.

Respectfully submitted,

/Daniel R. McClure/

Daniel R. McClure
Reg. No. 38,962

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, L.L.P.
100 Galleria Parkway NW
Suite 1750
Atlanta, Georgia 30339
(770) 933-9500